

Computational complexity analysis for Monte Carlo approximations of classically scaled population processes

David F. Anderson*, Desmond J. Higham†, Yu Sun‡

December 8, 2015

Abstract

We analyze and compare the computational complexity of different simulation strategies for Monte Carlo in the setting of classically scaled population processes. This setting includes stochastically modeled biochemical systems. We consider the task of approximating the expected value of some function of the state of the system at a fixed time point. We study the use of standard Monte Carlo when samples are produced by exact simulation and by approximation with tau-leaping or an Euler-Maruyama discretization of a diffusion approximation. Appropriate modifications of recently proposed multilevel Monte Carlo algorithms are also studied for the tau-leaping and Euler-Maruyama approaches. In order to quantify computational complexity in a tractable yet meaningful manner, we consider a parameterization that, in the mass action chemical kinetics setting, corresponds to the classical system size scaling. We then introduce a novel asymptotic regime where the required accuracy is a function of the model scaling parameter. Our new analysis shows that for this particular scaling a diffusion approximation offers little from a computational standpoint. Instead, we find that multilevel tau-leaping, which combines exact and tau-leaped samples, is the most promising method. In particular, multilevel tau-leaping provides an unbiased estimate and, up to a logarithm factor, is as efficient as a diffusion approximation combined with multilevel Monte Carlo. Computational experiments confirm the effectiveness of the multilevel tau-leaping approach.

1 Introduction

For some large $N_0 > 0$ we consider a continuous time Markov chain satisfying the stochastic equation

$$X^{N_0}(t) = X^{N_0}(0) + \sum_{k=1}^K \frac{1}{N_0} Y_k \left(N_0 \int_0^t \lambda_k(X^{N_0}(s)) ds \right) \zeta_k, \quad (1)$$

*Department of Mathematics, University of Wisconsin, Madison, USA. anderson@math.wisc.edu, grant support from NSF-DMS-1318832 and Army Research Office W911NF-14-1-0401.

†Department of Mathematics and Statistics, University of Strathclyde, UK. d.j.higham@maths.strath.ac.uk, grant support from a Wolfson/Royal Society Research Merit Award and an EPSRC Fellowship with reference EP/M00158X/1.

‡Department of Mathematics, University of Wisconsin, Madison, USA. ysun@math.wisc.edu.

where $X^{N_0}(t) \in \mathbb{R}^d$, $K < \infty$, the Y_k are independent unit Poisson processes and, for each k , $\zeta_k \in \mathbb{R}^d$ and $\lambda_k : \mathbb{R}^d \rightarrow \mathbb{R}_{\geq 0}$ are Lipschitz continuous with bounded second derivatives. We further assume that $\lambda_k(X^N(0)) = O(1)$. This particular scaling choice is often termed the “classical scaling” [8, 25, 26]. We consider the task of numerically approximating $\mathbb{E}[f(X^{N_0}(t))]$, in the sense of confidence intervals, to some fixed tolerance $\varepsilon_0 < 1$, where f satisfies mild regularity conditions.

The class of models of the form (1) satisfying the above assumptions has a long history in terms of modelling [10, 11, 12, 28], analysis [8, 25, 26] and computation [17, 18]. The framework covers many application areas, including population dynamics [29], queueing theory [30], and several branches of physics [13]. In recent years, chemical and biochemical kinetics models in systems biology [31] have been the driving force behind a resurgence of activity in algorithmic developments, including tau-leaping [19] and its multilevel extension [3, 4]. In this setting, the parameter N_0 in (1) can represent Avogadro’s number multiplied by the volume, and in this classical scaling, species are measured in moles per liter. More generally, however, N_0 can just be considered a large number, often of the order 100s or 1000s. Our aim here is to present a unified computational complexity analysis for a range of Monte Carlo based methods. This allows us to make what we believe are the first concrete conclusions pertaining to the relative merits of current methods in a practically relevant asymptotic regime. Of particular note is that our analysis suggests that a diffusion approximation offers very few advantages from a computational standpoint.

In section 2, we discuss some of the issues involved in quantifying computational complexity in the present setting, and introduce a novel scaling regime in which clear-cut comparisons can be made. Further, a high-level summary of our main conclusions is presented. In section 3, we summarize two relevant methods for approximating the model (1): the tau-leap discretization method, and the Langevin or diffusion approximation. In section 4, we quantify the computational complexity of using exact simulation, tau-leaping, and simulation of the diffusion equation with standard Monte Carlo for approximating $E[f(X^{N_0}(t))]$ to a desired tolerance. Further, in subsection 4.2 we review the more recent multilevel methods and quantify the benefits of their use in both the tau-leaping and diffusion scenarios. In section 5, we provide numerical examples demonstrating our main conclusions. In section 6, we close with some brief conclusions.

2 Scaling regime and summary of results

In order to motivate our analysis and computations, we begin with a brief, high-level, overview. In particular, we discuss the entries in Table 1, which summarizes the key results of this work. Full details are given later in the manuscript, however we point out here that the terms in Table 1 include assumptions on the variances of the constituent processes that will be made clear, and proven in a wide variety of cases, in the subsequent sections.

A natural approach to approximate the desired expectation is to simulate paths exactly, for example with the stochastic simulation algorithm [17, 18] or the next reaction method [1, 14], in order to obtain independent sample paths $\{X_{[i]}^{N_0}\}_{i=1}^n$ that can be combined into a

sample average

$$\hat{\mu}_n = \frac{1}{n} \sum_{i=1}^n f(X_{[i]}^{N_0}(t)). \quad (2)$$

This becomes problematic if the cost of each sample path is high—to follow a path exactly we must take account of each individual transition in the process. This is a serious issue when many jumps take place, which is the case when N_0 is large.

The essence of the Euler tau-leaping approach is to fix the system intensities over time intervals of length h , and thereby only require the generation of K Poisson random variables per time interval [19]. In order to analyse the benefit of tau-leaping, and related methods, Anderson, Ganguly, and Kurtz [2] considered the limit $N \rightarrow \infty$ and $h \rightarrow 0$ with $N = h^{-\beta}$ for some $\beta > 0$. To see why such a limit is useful we note two facts:

- If, instead, we allow $N \rightarrow \infty$ with h fixed, then the stochastic fluctuations become negligible [7, 25]. In this *thermodynamic limit* the model reduces to a deterministic ODE, so a simple deterministic numerical method could be used.
- If, instead, we allow $h \rightarrow 0$ with N fixed then tau-leaping becomes arbitrarily inefficient. The “empty” waiting times between reactions, which have nonzero expected values, are being needlessly refined by the discretization method.

The relation $N = h^{-\beta}$ brings together the large system size effect (where exact simulation is expensive and tau-leaping offers a computational advantage) with the the small h effect (where the accuracy of tau-leaping can be analysed). This gives a realistic setting where the benefits of tau-leaping can be quantified. It may then be shown [2, Theorem 4.1] that the bias arising from Euler tau-leaping is $O(h) = O(N^{-\beta})$.

For our purposes, rather than the step size h of a particular approximate method, it is more natural to work in terms of the system size, N_0 . First, let $\alpha > 0$ satisfy $\varepsilon_0 = N_0^{-\alpha}$. Next, consider the following family of models parameterized by $N \geq N_0$,

$$X^N(t) = X^N(0) + \sum_{k=1}^K \frac{1}{N} Y_k \left(N \int_0^t \lambda_k(X^N(s)) ds \right) \zeta_k. \quad (3)$$

We will study the asymptotic behavior, as $N \rightarrow \infty$, of the computational complexity required to approximate $\mathbb{E}[f(X^N(t))]$ to a tolerance of

$$\varepsilon_N = N^{-\alpha}. \quad (4)$$

We emphasize at this stage that we are no longer studying a fixed model. Instead we look at the family of models (3) parameterized through the system size N , and consider the limit, as $N \rightarrow \infty$, of the computational complexity of the different methods under the accuracy requirement (4). The computed results then tell us, to leading order, the costs associated with solving our fixed problem (1) with accuracy requirement $N_0^{-\alpha}$.

Monte Carlo method	Computational complexity	unbiased?	Most efficient
MC + exact simulation	$O(N^{2\alpha} + N)$	Yes	Never
MC + tau-leaping	$O(N^{3\alpha-1} + N^\alpha)$	No	Never
MC + midpt. or trap. tau-leap	$O(N^{2.5\alpha-1} + N^{\alpha/2})$	No	$\frac{1}{2} < \alpha \leq \frac{2}{3}$
MC + Euler for diff. approx.	$O(N^{3\alpha-1} + N^\alpha)$	No	Never
MLMC + E-M for diff. approx.	$O(N^{2\alpha-1} + N^\alpha)$	No	$\alpha \geq \frac{2}{3}$
biased MLMC tau-leaping	$O(N^{2\alpha-1} \log(N)^2 + N^\alpha)$	No	$\alpha \geq \frac{2}{3}$
unbiased MLMC tau-leaping	$O(N^{2\alpha-1} \log(N)^2 + N)$	Yes	$\alpha \geq 1$

Table 1: Computational cost for different Monte Carlo methods, as $N \rightarrow \infty$. The final column indicates when each method is most efficient, in terms of the parameter α , up to factors involving logarithms.

2.1 Summary of results

To quantify computational complexity, we define the “cost-per-path” to be the number of random variables generated in the simulation of a single path. We argue in section 4.1.1 that the expected cost-per-path of exact simulation scales linearly with N . Tau-leaping, by contrast, has a cost-per-path of $O(h^{-1})$, and we need $h \sim \varepsilon_N$ to get a small enough bias in our standard Monte Carlo approximation. Since the random variables of interest, $f(X^N(t))$, generally have a variance that scales like $O(N^{-1})$, see [4], the costs of standard Monte Carlo methods then become:

Monte Carlo plus exact simulation: $O(N^{-1}\varepsilon_N^{-2} + 1)$ paths at a cost of $O(N)$ per path, totaling a computational complexity of $O(\varepsilon_N^{-2} + N)$ or $O(N^{2\alpha} + N)$.

Monte Carlo plus tau-leaping: $O(N^{-1}\varepsilon_N^{-2} + 1)$ paths at a cost of $O(\varepsilon_N^{-1})$ per path, totaling a computational complexity of $O(N^{-1}\varepsilon_N^{-3} + \varepsilon_N^{-1})$ or $O(N^{3\alpha-1} + N^\alpha)$,

as summarized in the first two rows of Table 1. Note that the “+1” terms above account for the requirement that we cannot generate less than one path. In this regime, we see that tau-leaping is beneficial for $\alpha < 1$. This makes sense intuitively. If we ask for too much accuracy relative to the system size ($\alpha > 1$ in (4)) then tau-leaping’s built-in bias outweighs its cheapness, or, equivalently, the required stepsize is so small that tau-leaping works harder than exact simulation.

Higher order alternatives to the original tau-leaping method [19] are available. For example, a mid-point discretization [2, Theorem 4.2] or a trapezoidal method [6] both achieve $O(h^2) = O(N^{-2\alpha})$ bias, which allows the overall complexity to be reduced to $O(N^{2.5\alpha-1} + N^{\alpha/2})$ in this regime. This makes midpoint or trapezoidal tau-leaping beneficial as compared with exact simulation for $\alpha < 2$.

As an alternative to tau-leap discretizations, we could replace the continuous-time Markov chain by a diffusion approximation and use a numerical stochastic differential equation (SDE) simulation method to generate approximate paths. Even assuming that such an approximation produces an unbiased and consistent estimator (which is not, in general, the case) we find

that an Euler method with standard Monte Carlo gives the same cost as basic tau-leaping; see section 4.1.3.

Multilevel versions of Monte Carlo (MLMC) have been developed in both the Euler + diffusion setting [15, 16] and tau-leaping setting [3, 4]. Under mild assumptions on f and the stochastic system, we find that a multilevel scheme can reduce the standard Monte Carlo Euler/diffusion cost from $O(N^{3\alpha-1} + N^\alpha)$ to $O(N^{2\alpha-1} \log(N)^2 + N^\alpha)$. Similarly, the biased version of the multilevel algorithm associated with tau-leaping [3] can also give a complexity of $O(N^{2\alpha-1} \log(N)^2 + N^\alpha)$, and this is achieved without perturbing the basic model by taking a diffusion approximation. Further, an *unbiased* version of the multilevel algorithm yields a complexity of

$$O(N^{2\alpha-1} \log(N)^2 + N)$$

which is equivalent to the biased version when $\alpha \geq 1$.

We also mention that a crude and inexpensive approximation to the required expected value can be computed by simply simulating the deterministic mass action ODE approximation to (1), which is often referred to as the reaction rate equation. Ignoring all fluctuations in this manner, using a perturbed model that no longer depends upon N produces an approximation to the mean that has an error of $O(1/\sqrt{N})$, meeting the required $O(N^{-\alpha})$ accuracy requirement when $\alpha \leq \frac{1}{2}$. In this case, solving the deterministic ODE with a p th order numerical method scheme (such as a Runge–Kutta method) requires a stepsize with $h^p = \varepsilon_N$, so that $h = N^{-\alpha/p}$, giving an asymptotic computational cost of $O(h^{-1})$; that is, $O(N^{\alpha/p})$. Since the order p can be made arbitrarily large, we see that this approach has negligible cost. For this reason, we view $\alpha = \frac{1}{2}$ as a natural cut-off in the relationship (4); we are not concerned with $\alpha \leq \frac{1}{2}$ since in this regime the requested level of accuracy does not require fluctuations to be respected.

In addition to the asymptotic complexity counts in Table 1, another important feature of a method is the availability of computable *a posteriori* confidence interval information. As indicated in the table, two of the methods considered here, exact simulation with Monte Carlo and an appropriately constructed multilevel tau-leaping, are unbiased. The sample mean, accompanied by an estimate of the overall variance, can then be delivered with a computable confidence interval. By contrast, the remaining methods in the table are biased: tau-leaping and Euler–Maruyama introduce discretization errors and the diffusion approximation perturbs the underlying model. Although the asymptotic leading order of these biases can be estimated, useful *a posteriori* upper bounds cannot be computed straightforwardly in general, making these approaches much less attractive for reliably achieving a target accuracy.

Based on the range of methods analysed here in an asymptotic regime that couples system size and target accuracy, two key messages are

- even assuming there is no bias to the underlying model, simulating at the level of the diffusion approximation is only marginally advantageous,
- tau-leaping can offer advantages over exact simulation, and an appropriately designed version of multilevel tau-leaping (which combines exact and tau-leaped samples) offers an unbiased method that is efficient over a wide range of accuracy requirements.

3 Approximation methods

In this section, we briefly review two alternatives to exact simulation of (3).

3.1 Tau-Leaping

Tau-leaping [19] is a computational method that generates Euler-style approximate paths for the continuous-time Markov chain (3). The basic idea is to hold the intensity functions fixed over a time interval $[t_n, t_n + h]$ at the values $\lambda_k(X^N(t_n))$, where $X^N(t_n)$ is the state of the system at time t_n , and, under this assumption, compute the number of times each reaction takes place over this period. As the waiting times for the reactions are exponentially distributed, this leads to the following algorithm, which simulates up to a time of $T > 0$. For $x \geq 0$ we will write $\text{Poisson}(x)$ to denote a sample from the Poisson distribution with parameter x , with all such samples being independent of each other and of all other sources of randomness used.

Algorithm 1 (Euler tau-leaping). Fix $h > 0$. Set $Z_h^N(0) = x_0$, $t_0 = 0$, $n = 0$ and repeat the following until $t_{n+1} = T$:

- (i) Set $t_{n+1} = t_n + h$. If $t_{n+1} \geq T$, set $t_{n+1} = T$ and $h = T - t_n$.
- (ii) For each k , let $\Lambda_k = \text{Poisson}(\lambda_k(Z_h^N(t_n))h)$.
- (iii) Set $Z_h^N(t_{n+1}) = Z_h^N(t_n) + \sum_k \Lambda_k \zeta_k$.
- (iv) Set $n \leftarrow n + 1$.

Analogously to (3), a path-wise representation of Euler tau-leaping defined for all $t \geq 0$ can be given through a random time change of Poisson processes:

$$Z_h^N(t) = Z_h^N(0) + \sum_k Y_k \left(\int_0^t \lambda_k(Z_h^N(\eta_h(s))) ds \right) \zeta_k, \quad (5)$$

where the Y_k are as before, and $\eta_h(s) \stackrel{\text{def}}{=} \left\lfloor \frac{s}{h} \right\rfloor h$. Thus, $Z_h^N(\eta_h(s)) = Z_h^N(t_n)$ if $t_n \leq s < t_{n+1}$.

3.2 Diffusion approximation

The tau-leaping algorithm utilizes a time-stepping method to directly approximate the underlying model (3). Alternatively, a diffusion approximation arises by perturbing the underlying model into one which can be discretized more efficiently.

Define the function F via

$$F(x) = \sum_k \lambda_k(x) \zeta_k.$$

By the functional central limit theorem,

$$\frac{1}{\sqrt{N}} [Y_k(Nu) - Nu] \approx W_k(u), \quad (6)$$

where W_k is a standard Brownian motion. Applying (6) to (3) yields

$$X^N(t) \approx X^N(0) + \int_0^t F(X^N(s))ds + \sum_k \frac{1}{\sqrt{N}} W_k \left(\int_0^t \lambda_k(X^N(s))ds \right) \zeta_k,$$

where the W_k are independent standard Brownian motions. This implies that X^N can be approximated by the process D^N satisfying

$$D^N(t) = D^N(0) + \int_0^t F(D^N(s))ds + \sum_k \frac{1}{\sqrt{N}} W_k \left(\int_0^t \lambda_k(D^N(s))ds \right) \zeta_k,$$

where $D^N(0) = X^N(0)$. An equivalent, and more prevalent, way to represent D^N is via the Itô representation

$$D^N(t) = D^N(0) + \int_0^t F(D^N(s))ds + \sum_k \frac{1}{\sqrt{N}} \zeta_k \int_0^t \sqrt{\lambda_k(D^N(s))} dW_k(s), \quad (7)$$

which is often written in the differential form

$$dD^N(t) = F(D^N(t))dt + \sum_k \frac{1}{\sqrt{N}} \zeta_k \sqrt{\lambda_k(D^N(s))} dW_k(s). \quad (8)$$

The SDE system (8) is known as a *Langevin* approximation in the biology and chemistry literature, and a *diffusion* approximation in probability [8, 31]. We note the following points.

- The diffusion coefficient, often termed the “noise” in the system, is $O(\frac{1}{\sqrt{N}})$, and hence, in our setting is small relative to the drift.
- The diffusion coefficient involves square roots. Hence, it is critical that the choice of intensity functions λ_k only take values in $\mathbb{R}_{\geq 0}$ on the domain of the solution. This is of particular importance in the population process setting where the solutions of the underlying model (3) naturally satisfy a non negativity constraint whereas the SDE solution paths cannot be guaranteed to remain non-negative in general. Therefore, in this setting of population processes one reasonable representation, of many, would be

$$dD^N(t) = F(D^N(t))dt + \sum_k \frac{1}{\sqrt{N}} \zeta_k \sqrt{[\lambda_k(D^N(s))]^+} dW_k(s), \quad (9)$$

where $[x]^+ = \max\{x, 0\}$.

- The coefficients of the SDE are not globally Lipschitz in general, and hence standard convergence theory for numerical methods, such as that in [24], is not applicable. Examples of nonlinear SDEs for which standard Monte Carlo and multilevel Monte Carlo, when combined with Euler-Maruyama discretization, fail to produce a convergent algorithm have been pointed out in the literature [21, 20]. The question of which classes of reaction system lead to well-defined SDEs and which discretizations converge at the traditional rate therefore remains open.

In this work, to get a feel for the best possible computational complexity that can arise from the Langevin approximation, we will study the case where the function F is linear. In this case the first two moments of the diffusion approximation match those of the underlying continuous time Markov chain exactly [8]. We will find that even in this idealized light, the asymptotic computational complexity of Euler-Maruyama on a diffusion approximation combined with either a standard or a multilevel implementation is only marginally better than the corresponding computational complexity bounds for multilevel tau-leaping. In particular, they differ only in a log factor. We note that we only make the linearity assumption on the diffusion model.

4 Complexity analysis

In this section we establish the results given in Table 1. In subsection 4.1, we derive the first four rows, whereas in subsection 4.2 we discuss the multilevel framework and establish rows five, six, and seven.

4.1 Complexity analysis of standard Monte Carlo approaches

4.1.1 Exact Sampling and Monte Carlo

Suppose that we compute exact samples from the process $X^N(t)$. The intensity functions scale like $O(N)$ and the expected holding time between reactions is $O(N^{-1})$. Hence the number of system updates required to generate a single path is $O(N)$. Letting

$$\delta_N = \text{Var}(f(X^N(t)))$$

we require

$$n^{-1}\delta_N = O(\varepsilon_N^2) \implies n = O(\delta_N \varepsilon_N^{-2} + 1).$$

Thus, the total computational complexity of making the desired approximation is

$$O(nN) = O(\delta_N \varepsilon_N^{-2} N + N) = O(\delta_N N^{2\alpha+1} + N).$$

In [4] it is shown that $\text{Var}(X_i^N(t)) = O(N^{-1})$, giving $\delta_N = N^{-1}$ and an overall complexity of $O(N^{2\alpha} + N)$, as given in the first row of Table 1.

4.1.2 Tau-leaping and Monte Carlo

Suppose now that we use n paths of the tau-leaping process (5) to construct the Monte Carlo estimator $\hat{\mu}_n$ for $\mathbb{E}[f(X^N(t))]$. We note that

$$\mathbb{E}[f(X^N(t))] - \hat{\mu}_n = (\mathbb{E}[f(X^N(t))] - \mathbb{E}[f(Z_h^N(t))]) + (\mathbb{E}[f(Z_h^N(t))] - \hat{\mu}_n). \quad (10)$$

Tau-leaping was studied in [2] in a regime of the form $h = O(\varepsilon_N)$. By constraining $h = O(\varepsilon_N)$ we can apply [2, Theorem 4.1] to give

$$\mathbb{E}[f(X^N(t))] - \mathbb{E}[f(Z_h^N(t))] = O(h) = O(\varepsilon_N),$$

for a wide class of functionals f , as required for the bias. Letting

$$\delta_{N,h} = \text{Var}(f(Z_h^N(t)))$$

we again require $n = O(\delta_{N,h}\varepsilon_N^{-2} + 1)$ to control the statistical error. Since there are $O(h^{-1})$ operations per path generation, the total computational complexity for making the desired approximation is

$$O(nh^{-1}) = O(\delta_{N,h}\varepsilon_N^{-3} + \varepsilon_N^{-1}).$$

We know from [4] that $\text{Var}(Z_{h,i}^N(t)) = O(N^{-1})$, giving an overall complexity of $O(N^{3\alpha-1} + N^\alpha)$, as reported in the second row of Table 1.

Weakly second order extensions to the tau-leaping method can lower the computational complexity dramatically. For example, if we use the midpoint tau-leaping process \mathcal{Z}_h^N from [2], we can set $h = \sqrt{\varepsilon_N}$ and still achieve a bias of $O(\varepsilon_N)$. In particular, from [2, Theorem 4.2], we have

$$\mathbb{E}[f(X^N(t))] - \mathbb{E}[F(\mathcal{Z}_h^N(t))] = O(h^2),$$

for a wide class of f . By similar methods as in [4], it can be shown that \mathcal{Z}_h^N has the same variance scaling as X_h^N and Z_h^N . Since we need $n = O(\delta_{N,h}\varepsilon_N^{-2} + 1)$ paths, the complexity is

$$O(n \cdot h^{-1}) = O(n \cdot \varepsilon_N^{-1/2}) = O(\delta_{N,h}\varepsilon_N^{-2.5} + \varepsilon_N^{-1/2}).$$

When $\delta_{N,h} = N^{-1}$ the resulting complexity is $O(N^{2.5\alpha-1} + N^{\alpha/2})$, as stated in the third row of Table 1. The same conclusion can also be drawn for the trapezoidal method in [6].

4.1.3 Diffusion approximation and Monte Carlo

In considering an Euler–Maruyama discretization of the diffusion approximation, it is important to keep in mind that we are studying a parameterized class of SDEs in the $N \rightarrow \infty$ limit, rather than a single SDE. However, because the drift remains $O(1)$, the weak error stays at the $O(h)$ level arising from the underlying deterministic Euler method; that is, Euler–Maruyama introduces a bias of $O(h)$, uniformly in N . So we need $h = O(\varepsilon_N)$ to keep the bias within our tolerance. Letting

$$\delta_{N,h} = \text{Var}(f(D_h^N(t))),$$

we require $n = O(\delta_{N,h}\varepsilon_N^{-2} + 1)$ to control the statistical error. Since there are $O(h^{-1})$ operations per path generation, the total computational complexity for making the desired approximation is

$$O(nh^{-1}) = O(\delta_{N,h}\varepsilon_N^{-3} + \varepsilon_N^{-1}).$$

Since by standard methods we can conclude $\text{Var}(D_{h,i}^N(t)) = O(N^{-1})$, we can again give an overall complexity of $O(N^{3\alpha-1} + N^\alpha)$, as reported in the fourth row of Table 1.

4.2 Multilevel Monte Carlo and complexity analysis

In this section we study multilevel Monte Carlo approaches and derive the results summarized in rows five, six, and seven of Table 1.

4.2.1 Multilevel Monte Carlo and Diffusion Approximation

Here we specify and analyze an Euler-based multilevel method for the diffusion approximation, following the original framework of Giles [16].

For some fixed $M > 1$ we let $h_\ell = T \cdot M^{-\ell}$ for $\ell \in \{0, \dots, L\}$. Reasonable choices for M include $M \in \{2, 3, 4, \dots, 7\}$, and L is determined below. Let $D_{h_\ell}^N$ denote the approximate process generated by Euler-Maruyama applied to (8) with a step size of h_ℓ . As mentioned in section 4.1.3, the discretization has weak order one for a large class of functionals f , so we set $h_L = \varepsilon_N$, giving $L = O(|\log(\varepsilon_N)|)$, so that the finest level achieves the required order of magnitude for the bias.

Noting that

$$E[f(D_{h_L}^N(t))] = E[f(D_{h_0}^N(t))] + \sum_{\ell=1}^L E[f(D_{h_\ell}^N(t)) - f(D_{h_{\ell-1}}^N(t))], \quad (11)$$

we use i as an index over sample paths and let

$$\widehat{Q}_0^N \stackrel{\text{def}}{=} \frac{1}{n_0} \sum_{i=1}^{n_0} f(D_{h_0, [i]}^N(t)), \quad \text{and} \quad \widehat{Q}_\ell^N \stackrel{\text{def}}{=} \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} (f(D_{h_\ell, [i]}^N(t)) - f(D_{h_{\ell-1}, [i]}^N(t))),$$

for $\ell = 1, \dots, L$, where n_0 and the different n_ℓ have yet to be determined. Note that the form of the estimator \widehat{Q}_ℓ^N above implies that the processes $D_{h_\ell}^N$ and $D_{h_{\ell-1}}^N$ will be *coupled*, or constructed on the same probability space. We consider here the case when $(D_{h_\ell}^N, D_{h_{\ell-1}}^N)$ are coupled in the usual way by using the same Brownian path in the generation of each of the marginal processes. Our (biased) estimator is then

$$\widehat{Q}^N \stackrel{\text{def}}{=} \widehat{Q}_0^N + \sum_{\ell=1}^L \widehat{Q}_\ell^N.$$

Set

$$\delta_{N,\ell} = \text{Var}(f(D_{h_\ell}^N(t)) - f(D_{h_{\ell-1}}^N(t))).$$

It is shown in [5] that $\delta_{N,\ell} = O(N^{-1}h_\ell^2 + N^{-2}h_\ell)$ under a wide array of conditions. Finally note that we also have $\text{Var}(f(D_0^N(t))) = O(N^{-1})$. In [5] it is shown that under these circumstances, the computational complexity required is $O(\varepsilon_N^{-2}N^{-1} + \varepsilon_N^{-1})$. In the regime (4) this translates to $O(N^{2\alpha-1} + N^\alpha)$, as reported in the fifth row of Table 1.

4.2.2 Multilevel Monte Carlo and tau-leaping

The use of multilevel Monte Carlo with tau-leaping for continuous-time Markov chains of the form considered here was proposed in [3], where effective algorithms were devised. Complexity results were given in a non-asymptotic multi-scale setting, with followup results in [4]. Our aim here is to customize the approach in the scaling regime (4) and thereby develop easily interpretable complexity bounds that allow straightforward comparison with other methods. In this section $Z_{h_\ell}^N$ denotes a tau-leaping process generated with a step-size of $h_\ell = T \cdot M^\ell$, for $\ell \in \{0, \dots, L\}$.

A major step in [3] was to show that a coupling technique used for analytical purposes in [2, 27] can also form the basis of a practical simulation algorithm. Letting $Y_{k,i}$, $i \in \{1, 2, 3\}$, denote independent, unit rate Poisson processes, we couple the exact and approximate tau-leaping processes in the following way,

$$\begin{aligned} X^N(t) = & X^N(0) + \sum_k \frac{1}{N} Y_{k,1} \left(N \int_0^t \lambda_k(X^N(s)) \wedge \lambda_k(Z_{h_L}^N(\eta_L(s))) ds \right) \zeta_k \\ & + \sum_k \frac{1}{N} Y_{k,2} \left(N \int_0^t [\lambda_k(X^N(s)) - \lambda_k(X^N(s)) \wedge \lambda_k(Z_{h_L}^N(\eta_L(s)))] ds \right) \zeta_k, \end{aligned} \quad (12)$$

$$\begin{aligned} Z_{h_L}^N(t) = & Z_{h_L}^N(0) + \sum_k \frac{1}{N} Y_{k,1} \left(N \int_0^t \lambda_k(X^N(s)) \wedge \lambda_k(Z_{h_L}^N(\eta_L(s))) ds \right) \zeta_k \\ & + \sum_k \frac{1}{N} Y_{k,3} \left(N \int_0^t [\lambda_k(Z_{h_L}^N(\eta_L(s))) - \lambda_k(X^N(s)) \wedge \lambda_k(Z_{h_L}^N(\eta_L(s)))] ds \right) \zeta_k, \end{aligned} \quad (13)$$

where $a \wedge b$ denotes $\min\{a, b\}$ and $\eta_L(s) = \lfloor s/h_L \rfloor h_L$. Sample paths of (12)–(13) can be generated with a natural extension of the next reaction method or Gillespie’s algorithm, see [3], and for $h_L \geq N^{-1}$ the complexity required for the generation of a realization $(X^N, Z_{h_L}^N)$ remains at the $O(N)$ level. The coupling of two approximate processes, $Z_{h_\ell}^N$ and $Z_{h_{\ell-1}}^N$, takes the form

$$\begin{aligned} Z_{h_\ell}^N(t) = & Z_{h_\ell}^N(0) + \sum_k \frac{1}{N} Y_{k,1} \left(N \int_0^t \lambda_k(Z_{h_\ell}^N(\eta_\ell(s))) \wedge \lambda_k(Z_{h_{\ell-1}}^N(\eta_{\ell-1}(s))) ds \right) \zeta_k \\ & + \sum_k \frac{1}{N} Y_{k,2} \left(N \int_0^t [\lambda_k(Z_{h_\ell}^N(\eta_\ell(s))) - \lambda_k(Z_{h_\ell}^N(\eta_\ell(s))) \wedge \lambda_k(Z_{h_{\ell-1}}^N(\eta_{\ell-1}(s)))] ds \right) \zeta_k, \end{aligned} \quad (14)$$

$$\begin{aligned} Z_{h_{\ell-1}}^N(t) = & Z_{h_{\ell-1}}^N(0) + \sum_k \frac{1}{N} Y_{k,1} \left(N \int_0^t \lambda_k(Z_{h_\ell}^N(\eta_\ell(s))) \wedge \lambda_k(Z_{h_{\ell-1}}^N(\eta_{\ell-1}(s))) ds \right) \zeta_k \\ & + \sum_k \frac{1}{N} Y_{k,3} \left(N \int_0^t [\lambda_k(Z_{h_{\ell-1}}^N(\eta_{\ell-1}(s))) - \lambda_k(Z_{h_\ell}^N(\eta_\ell(s))) \wedge \lambda_k(Z_{h_{\ell-1}}^N(\eta_{\ell-1}(s)))] ds \right) \zeta_k, \end{aligned} \quad (15)$$

where $\eta_\ell(s) \stackrel{\text{def}}{=} \lfloor s/h_\ell \rfloor h_\ell$. The pair (14)–(15) can be sampled at the same $O(h_\ell^{-1})$ cost as a single tau-leaping path, see [3].

For L as yet to be determined, and noting the identity

$$\mathbb{E}[f(X^N(t))] = \mathbb{E}[f(X^N(t)) - f(Z_L^N(t))] + \sum_{\ell=1}^L \mathbb{E}[f(Z_{h_\ell}^N(t)) - f(Z_{h_{\ell-1}}^N(t))] + \mathbb{E}[f(Z_{h_0}^N(t))], \quad (16)$$

we define estimators for the three terms above via

$$\begin{aligned}\widehat{Q}_E^N &\stackrel{\text{def}}{=} \frac{1}{n_E} \sum_{i=1}^{n_E} (f(X_{[i]}^N(t)) - f(Z_{h_L, [i]}^N(t))), \\ \widehat{Q}_\ell^N &\stackrel{\text{def}}{=} \frac{1}{n_\ell} \sum_{i=1}^{n_\ell} (f(Z_{h_\ell, [i]}^N(t)) - f(Z_{h_{\ell-1}, [i]}^N(t))), \quad \text{for } \ell \in \{1, \dots, L\}, \\ \widehat{Q}_0^N &\stackrel{\text{def}}{=} \frac{1}{n_0} \sum_{i=1}^{n_0} f(Z_{h_0, [i]}^N(t)),\end{aligned}\tag{17}$$

so that

$$\widehat{Q}^N \stackrel{\text{def}}{=} \widehat{Q}_E^N + \sum_{\ell=1}^L \widehat{Q}_\ell^N + \widehat{Q}_0^N \tag{18}$$

is an unbiased estimator for $\mathbb{E}[f(X^N(T))]$. Here, \widehat{Q}_E^N uses the coupling (12)–(13) between exact paths and tau-leaped paths of stepsize h_L , \widehat{Q}_ℓ^N uses the coupling (14)–(15) between tau-leaped paths of stepsizes h_ℓ and $h_{\ell-1}$, and \widehat{Q}_0^N involves single tau-leaped paths of stepsize h_0 . Note that the algorithm implicit in (18) produces an unbiased estimator, whereas the estimator is biased if \widehat{Q}_E^N is left off, as will sometimes be desirable. Hence, we will refer to estimator \widehat{Q}^N in (18) as the unbiased estimator, and will refer to

$$\widehat{Q}_B^N \stackrel{\text{def}}{=} \sum_{\ell=1}^L \widehat{Q}_\ell^N + \widehat{Q}_0^N \tag{19}$$

as the biased estimator. For both the biased and unbiased estimators, the number of paths at each level, n_0 , n_ℓ and n_E , will be chosen to ensure an overall estimator variance of $O(\varepsilon_N^2)$.

Continuing the analysis, we assume that for $\ell \geq 0$

$$\text{Var}(f(X^N(t)) - f(Z_{h_\ell}^N(t))) = O(\delta_{N,E}) \tag{20}$$

$$\text{Var}(f(Z_{h_\ell}^N(t)) - f(Z_{h_{\ell-1}}^N(t))) = O(\delta_{N,\ell}) \tag{21}$$

$$\text{Var}(f(Z_{h_0}^N(t))) = O(N^{-1}). \tag{22}$$

It is shown in [4] that $\delta_{N,\ell} = N^{-1}h_\ell$ for $\ell \geq 0$ under a wide array of circumstances, so for the remainder of this section we assume this scaling.

We consider the biased and unbiased versions of tau-leaping multilevel Monte Carlo separately.

Biased multilevel Monte Carlo tau-leaping

Here we consider the estimator \widehat{Q}_B^N defined in (19). We first note that $E[f(X^N(t))] - E[f(Z_{h_L}^N(t))] = O(h_L)$ for a large class of functionals f , see [2]. Hence, we begin by choosing $L = O(\log(1/\varepsilon_N)) = O(\log(N))$ in order to control the bias.

For $\ell \in \{1, \dots, L\}$, let C_ℓ be the number of random variables required to generate a single pair of coupled trajectories at level ℓ . Let C_0 be the computational complexity required to generate a single trajectory at the coarsest level. To find n_ℓ , $\ell \in \{0, \dots, L\}$, we solve the

following optimization problem, which ensures that the variance of \widehat{Q}_B^N is no greater than order ε_N^2 :

$$\underset{n_\ell}{\text{minimize}} \quad \sum_{\ell=0}^L n_\ell C_\ell, \quad (23)$$

$$\text{subject to} \quad \sum_{\ell=0}^L \frac{\delta_{N,\ell}}{n_\ell} = \varepsilon_N^2. \quad (24)$$

We use Lagrange multipliers. As $C_\ell = K \cdot h_\ell^{-1}$, for some fixed constant K , the optimization problem above is solved at solutions to

$$\nabla_{n_0, \dots, n_L, \lambda} \left(\sum_{\ell=0}^L n_\ell K \cdot h_\ell^{-1} + \lambda \left(\sum_{\ell=0}^L \frac{\delta_{N,\ell}}{n_\ell} - \varepsilon_N^2 \right) \right) = 0.$$

Taking derivatives with respect to n_ℓ and setting each derivative to zero yields,

$$n_\ell = \sqrt{\frac{\lambda}{K} \delta_{N,\ell} h_\ell}, \quad \text{for } \ell \in \{0, 1, 2, \dots, L\} \quad (25)$$

for some $\lambda \geq 0$. Plugging (25) into (24) gives us,

$$\sum_{\ell=0}^L \sqrt{\frac{\delta_{N,\ell}}{h_\ell}} = \sqrt{\frac{\lambda}{K}} \cdot \varepsilon_N^2 \quad (26)$$

and hence, by (21) and (22),

$$\sqrt{\frac{\lambda}{K}} = \sum_{\ell=0}^L \sqrt{\frac{\delta_{N,\ell}}{h_\ell}} \leq CL \varepsilon_N^{-2} N^{-1/2}, \quad (27)$$

where C is a constant. Noting that $L = O(\log(\varepsilon_N^{-1}))$, we have

$$\frac{\lambda}{K} = O\left(\varepsilon_N^{-4} \log(\varepsilon_N)^2 N^{-1}\right).$$

Plugging this back into (25), and recognizing that at least one path must be generated to achieve the desired accuracy, we find

$$n_\ell = O(\varepsilon_N^{-2} N^{-1} h_\ell L + 1).$$

Hence, the overall computational complexity is

$$\begin{aligned} \sum_{\ell=0}^L n_\ell K h_\ell^{-1} &= O\left(\sum_{\ell=0}^L \varepsilon_N^{-2} N^{-1} h_\ell L h_\ell^{-1} + \sum_{\ell=0}^L h_\ell^{-1}\right) \\ &= O\left(\varepsilon_N^{-2} N^{-1} \log(\varepsilon_N)^2 + \varepsilon_N^{-1}\right) \\ &= O\left(N^{2\alpha-1} \log(N)^2 + N^\alpha\right), \end{aligned}$$

recovering row six of Table 1.

Note that the computational complexity reported for this biased version of multilevel Monte Carlo tau-leaping is, up to logarithms, the same as that for multilevel Monte Carlo on the diffusion approximation. However, none of the generous assumptions we made for the diffusion approximation, including that $E[f(X^N(t))] = E[f(D^N(t))]$, were required.

Unbiased multilevel Monte Carlo tau-leaping

The first observation to make is that the telescoping sum (16) implies that the method which utilizes $E[f(X^N(t)) - f(Z_{h_L}^N(t))]$ at the finest level is unbiased for *any* choice of h_L . That is, we are no longer constrained to choose $L = O(|\log(\varepsilon_N)|)$.

Assume (20) holds with $\delta_{N,E} = N^{-1}h_L$, and that $h_L \geq N^{-1}$. Let C_E be the average number of random variables required to generate a single pair of the coupled exact and tau-leaped processes when the tau-leap discretization is h_L . We know $C_E = O(N + h_L^{-1}) = O(N)$. To determine n_ℓ and n_E , we still solve an optimization problem,

$$\underset{n_\ell}{\text{minimize}} \quad \sum_{\ell=0}^L n_\ell C_\ell + n_E C_E, \quad (28)$$

$$\text{subject to} \quad \sum_{\ell=0}^L \frac{\delta_{N,\ell}}{n_\ell} + \frac{\delta_{N,E}}{n_E} = \varepsilon_N^2. \quad (29)$$

Using Lagrange multipliers again, we obtain,

$$n_\ell = \sqrt{\frac{\lambda \delta_{N,\ell}}{C_\ell}} \quad \text{for } \ell \in \{0, 1, 2, \dots, L\} \quad (30)$$

and

$$n_E = \sqrt{\frac{\lambda \delta_{N,E}}{C_E}}. \quad (31)$$

Plugging back into (29) and noting $C_\ell = O(h_\ell^{-1})$ and $C_E = O(N)$ yields

$$\sqrt{\lambda} = \varepsilon_N^{-2} \left(\sum_{\ell=0}^L \sqrt{\delta_{N,\ell} C_\ell} + \sqrt{\delta_{N,E} C_E} \right) \leq C(L\varepsilon_N^{-2} N^{-1/2} + \varepsilon_N^{-2} \sqrt{h_L}). \quad (32)$$

Therefore, plugging (32) back into (30) and (31) and noting $n_\ell \geq 1$ and $n_E \geq 1$, we get

$$n_\ell = \sqrt{\frac{\lambda \delta_{N,\ell}}{C_\ell}} + 1 = O \left(\left(L\varepsilon_N^{-2} N^{-1} + \varepsilon_N^{-2} \sqrt{\frac{h_L}{N}} \right) h_\ell + 1 \right) \quad \text{for } \ell \in \{0, 1, 2, \dots, L\}$$

and

$$n_E = \sqrt{\frac{\lambda \delta_{N,E}}{C_E}} + 1 = O(L\varepsilon_N^{-2} N^{-3/2} h_L^{1/2} + \varepsilon_N^{-2} N^{-1} h_L + 1). \quad (33)$$

As a result the total complexity is

$$\begin{aligned}
g(h_L) &= O(\varepsilon_N^{-2} N^{-1} L^2 + \varepsilon_N^{-2} \sqrt{\frac{h_L}{N}} L + h_L^{-1} + \varepsilon_N^{-2} \sqrt{\frac{h_L}{N}} L + \varepsilon_N^{-2} h_L + N) \\
&\leq O(\varepsilon_N^{-2} N^{-1} L^2 + 2\varepsilon_N^{-2} \sqrt{\frac{h_L}{N}} L + \varepsilon_N^{-2} h_L + 2N) && (\text{since } h_L^{-1} \leq N) \\
&= O(2\varepsilon_N^{-2} N^{-1} L^2 + 2\varepsilon_N^{-2} h_L + 2N). && (\text{using that } 2ab \leq a^2 + b^2)
\end{aligned}$$

It is relatively easy to show that the last line above is minimized at

$$h_L = \frac{2}{N} \text{LambertW}\left(\frac{N}{2}\right) \approx \frac{2}{N} \log\left(\frac{N}{2}\right). \quad (34)$$

Hence, taking $h_L = O(N^{-1} \log(N))$, we have $\log(h_L)^2 = O(\log(N)^2)$ and this method achieves a total computational complexity of leading order

$$O(\varepsilon_N^{-2} N^{-1} \log(N)^2 + \varepsilon_N^{-2} N^{-1} \log(N) + N) = O(\varepsilon_N^{-2} N^{-1} \log(N)^2 + N) = O(N^{2\alpha-1} \log(N)^2 + N),$$

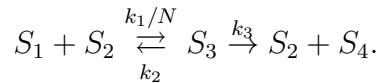
as reported in the last row of Table 1.

Note here that if we choose $h_L = \frac{1}{N}$ we get the same order of magnitude for the computational complexity. However the h_L in (34) is the optimized solution, meaning the leading order constant should be better and we will see this in Figure 3 and Figure 4 in the next section.

5 Computational results

In this section we provide numerical evidence for the sharpness of the computational complexity analyses provided in Table 1. We will measure complexity by total number of random variables utilized. We emphasize that these experiments use extreme parameter choices solely for the purpose of testing the sharpness of the delicate asymptotic bounds.

Example 1. We consider the classically scaled stochastic model for the following reaction network (see [8])



Letting $X_i(t)$ give the number of molecules of species S_i at time t , and letting $X^N(t) =$

$X(t)/N$, the stochastic equations are

$$\begin{aligned} X^N(t) = X^N(0) &+ \frac{1}{N} Y_1 \left(N k_1 \int_0^t X_1^N(s) X_2^N(s) ds \right) \begin{bmatrix} -1 \\ -1 \\ 1 \\ 0 \end{bmatrix} \\ &+ \frac{1}{N} Y_2 \left(N k_2 \int_0^t X_3^N(s) ds \right) \begin{bmatrix} 1 \\ 1 \\ -1 \\ 0 \end{bmatrix} \\ &+ \frac{1}{N} Y_3 \left(N k_3 \int_0^t X_3^N(s) ds \right) \begin{bmatrix} 0 \\ 1 \\ -1 \\ 1 \end{bmatrix}, \end{aligned}$$

where we assume $X^N(0) = O(1)$. We implemented different Monte Carlo simulation methods for the estimation of $\mathbb{E}[X_1^N(T)]$ to an accuracy of $\varepsilon_N = N^{-\alpha}$ for both $\alpha = 1$ and $\alpha = 5/4$. Specifically, for each of the order one methods we chose a step size of size $h = \varepsilon_N$ and required the variance of the estimator to be ε_N^2 . For midpoint tau-leaping, which has a weak order of two, we chose $h = \sqrt{\varepsilon_N}$. For the unbiased multilevel Monte Carlo method we chose the finest time-step according to (34). We do not provide estimates for Monte Carlo combined with exact simulation as those computations were too intensive to complete to the target accuracy.

For our numerical example we chose $T = 1$ and $X(0) = \lceil N \cdot [0.2, 0.2, 0.2, 0.2]^T \rceil$ with $X^N(0) = X(0)/N$. Finally, we chose $k_1 = k_2 = k_3 = 1$ as our rate constants. In Figure 1, we provide log-log plots of the computational complexity required to solve this problem for the different Monte Carlo methods to an accuracy of $\varepsilon_N = N^{-1}$, for each of

$$N \in \{2^{13}, 2^{14}, 2^{15}, 2^{16}, 2^{17}\}.$$

In Figure 2, we provide log-log plots for the computational complexity required to solve this problem for the different methods to an accuracy of $\varepsilon_N = N^{-\frac{5}{4}}$, for each of

$$N \in \{2^9, 2^{10}, 2^{11}, 2^{12}, 2^{13}\}.$$

Tables 2 and 3 provide the estimator variances for the different Monte Carlo methods with $\varepsilon_N = N^{-1}$ and $\varepsilon_N = N^{-\frac{5}{4}}$, respectively. The top line provides the target variances.

The specifics of the implementations and results for the different Monte Carlo methods are detailed below.

Diffusion Approximation plus Monte Carlo. We took a time step of size $h = \varepsilon_N$ to generate our independent samples. See Figure 1, where the best fit line is $y = 1.94x - 0.88$, and Figure 2, where the best fit line is $y = 2.73x - 1.37$, which are consistent with Table 1.

Monte Carlo Tau-Leaping. We took a time step of size $h = \varepsilon_N$ to generate our independent samples. See Figure 1, where the best fit line is $y = 1.96x - 1.02$, and Figure 2, where the best fit line is $y = 2.76x - 1.63$, which are consistent with Table 1.

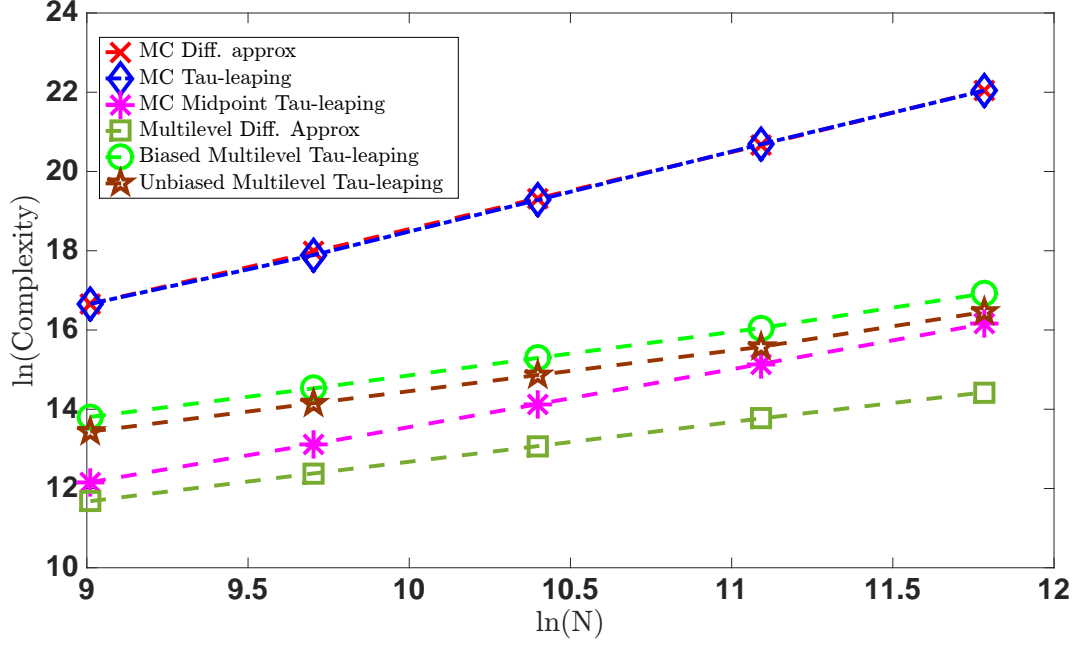


Figure 1: Log-log plots of the computational complexity for the different Monte Carlo methods with varying $N \in \{2^{13}, 2^{14}, 2^{15}, 2^{16}, 2^{17}\}$ and $\varepsilon_N = N^{-1}$.

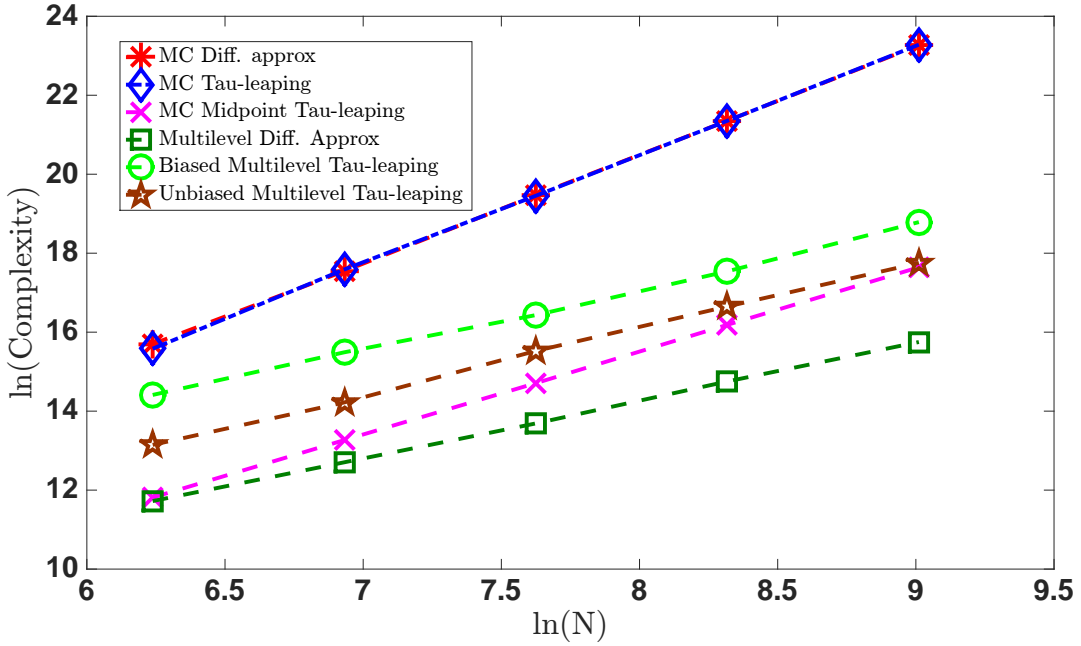


Figure 2: Log-log plots of the computational complexity for the different Monte Carlo methods with varying $N \in \{2^9, 2^{10}, 2^{11}, 2^{12}, 2^{13}\}$ and $\varepsilon_N = N^{-\frac{5}{4}}$.

Method	estimator standard deviations
	$2^{-13}, 2^{-14}, 2^{-15}, 2^{-16}, 2^{-17}$
MC and Diff. approx	$2^{-13.10}, 2^{-14.02}, 2^{-15.02}, 2^{-16.01}, 2^{-17.00}$
MC and Tau-leaping	$2^{-13.09}, 2^{-14.01}, 2^{-15.01}, 2^{-16.01}, 2^{-17.00}$
MC and Midpoint Tau-leaping	$2^{-13.09}, 2^{-14.04}, 2^{-15.03}, 2^{-16.00}, 2^{-17.01}$
Multilevel Diff. approx	$2^{-13.20}, 2^{-14.15}, 2^{-15.11}, 2^{-16.09}, 2^{-17.07}$
Biased Multilevel Tau-leaping	$2^{-13.44}, 2^{-14.39}, 2^{-15.39}, 2^{-16.38}, 2^{-17.32}$
Unbiased Multilevel Tau-leaping	$2^{-13.29}, 2^{-14.28}, 2^{-15.26}, 2^{-16.21}, 2^{-17.18}$

Table 2: Actual estimator variances when $\varepsilon_N = N^{-1}$.

Method	estimator standard deviations
$\varepsilon_N = N^{-\frac{5}{4}}$	$2^{-11.25}, 2^{-12.50}, 2^{-13.75}, 2^{-15.00}, 2^{-16.25}$
MC and Diff. approx	$2^{-11.27}, 2^{-12.51}, 2^{-13.75}, 2^{-15.00}, 2^{-16.25}$
MC and Tau-leaping	$2^{-11.26}, 2^{-12.52}, 2^{-13.76}, 2^{-15.00}, 2^{-16.25}$
MC and Midpoint Tau-leaping	$2^{-11.26}, 2^{-12.52}, 2^{-13.76}, 2^{-15.00}, 2^{-16.25}$
Multilevel Diff. approx	$2^{-11.46}, 2^{-12.63}, 2^{-13.85}, 2^{-15.06}, 2^{-16.29}$
Biased Multilevel Tau-leaping	$2^{-11.62}, 2^{-12.81}, 2^{-13.99}, 2^{-15.19}, 2^{-16.41}$
Unbiased Multilevel Tau-leaping	$2^{-11.34}, 2^{-12.57}, 2^{-13.79}, 2^{-15.03}, 2^{-16.26}$

Table 3: Actual estimator variances when $\varepsilon_N = N^{-5/4}$.

Monte Carlo Midpoint Tau-Leaping. We took a time step of size $h = \sqrt{\varepsilon_N}$. See Figure 1, where the best fit line is $y = 1.44 - 0.86$, and Figure 2, where the best fit line is $y = 2.10x - 3.53$, which are consistent with Table 1.

Our implementation of the multilevel methods proceeded as follows. We chose $h_\ell = 2^{-\ell}$ and for $\varepsilon_N > 0$ we fixed $h_L = \varepsilon_N$ and $L = \lceil \log(h_L)/\log(2) \rceil$ for the biased methods. For each level we generated N_0 independent sample trajectories in order to estimate $\delta_{N,\ell}$, as defined in section 3. Then we selected

$$n_\ell = \left\lceil \varepsilon_N^{-2} \sqrt{\delta_{N,\ell} h_\ell} \sum_{j=0}^L \sqrt{\frac{\delta_{N,j}}{h_j}} \right\rceil + 1, \quad \text{for } \ell \in \{0, 1, 2, \dots, L\},$$

to ensure the overall variance is below the target ε_N^2 .

Multi-Level Monte Carlo Diffusion Approximation We used $N_0 = 400$ for our pre-calculation of the variances. See Figure 1, where the best fit line is $y = 0.99x + 2.75$, and Figure 2, where the best fit line is $y = 1.45x + 2.61$, which are consistent with Table 1.

Multi-Level Monte Carlo Tau-Leaping. We used $N_0 = 100$ for our pre-calculation of the variances. See Figure 1, where the best fit line is $y = 1.12x + 3.70$, and Figure 2, where the best fit line is $y = 1.56x + 4.64$, which are consistent with Table 1.

Unbiased Tau-leaping multilevel Monte Carlo. For our implementation of unbiased multilevel tau-leaping, we set $h_L = \frac{2}{N} \text{LambertW}\left(\frac{N}{2}\right)$ and $L = \lceil \log(h_L)/\log(2) \rceil$. For each

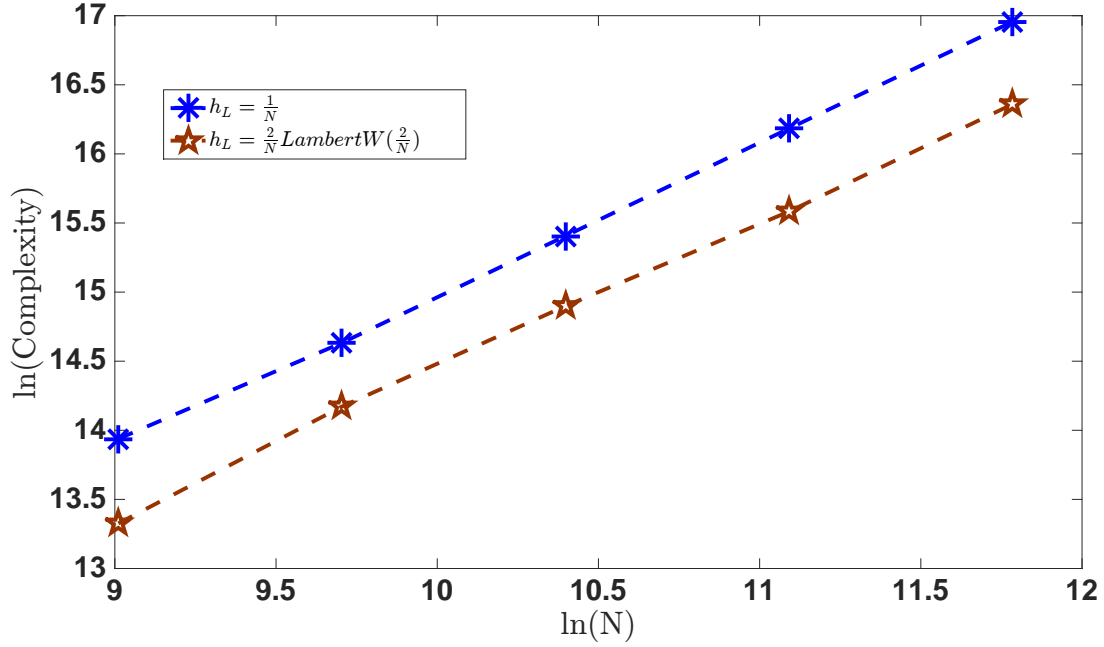


Figure 3: Complexity comparison of unbiased multilevel Monte Carlo tau-leaping when $h_L = \frac{1}{N}$ and $h_L = \frac{2}{N}\text{LambertW}(\frac{2}{N})$, with $\varepsilon_N = N^{-1}$.

level we utilized $N_0 = 100$ independent sample trajectories in order to estimate $\delta_{N,\ell}, C_\ell$ and $\delta_{N,E}, C_E$, as defined in section 3. Then we then selected

$$n_\ell = \left\lceil \varepsilon_N^{-2} \sqrt{\frac{\delta_{N,\ell}}{C_\ell}} \left(\sum_{\ell=0}^L \sqrt{\delta_{N,\ell} C_\ell} + \sqrt{\delta_{N,E} C_E} \right) \right\rceil + 1, \quad \text{for } \ell \in \{0, 1, 2, \dots, L\},$$

and

$$n_E = \left\lceil \varepsilon_N^{-2} \sqrt{\frac{\delta_{N,E}}{C_E}} \left(\sum_{\ell=0}^L \sqrt{\delta_{N,\ell} C_\ell} + \sqrt{\delta_{N,E} C_E} \right) \right\rceil + 1,$$

to ensure the overall estimator variance is below our target ε_N^2 . See Figure 1, where the best fit line is $y = 1.08x + 3.71$, and Figure 2, where the best fit line is $y = 1.68x + 2.65$, which are consistent with Table 1.

We also used the unbiased tau-leaping multilevel Monte Carlo method with $h_L = N^{-1}$ to estimate $\mathbb{E}[X_1(1)]$ to accuracy $\varepsilon_N = N^{-\alpha}$, for both $\alpha = 1$ and $\alpha = 5/4$. See Figures 3 and 4 for log-log plots of the required complexity when $h_L = N^{-1}$ and $h_L = (\frac{2}{N})\text{LambertW}(\frac{N}{2})$. As predicted in section 4.2.2, the complexity required when $h_L = (\frac{2}{N})\text{LambertW}(\frac{N}{2})$ is lower by some constant factor.

6 Conclusions

Many researchers have observed in practice that approximation methods can lead to computational efficiency, relative to exact path simulation. However, *meaningful, rigorous justifica-*

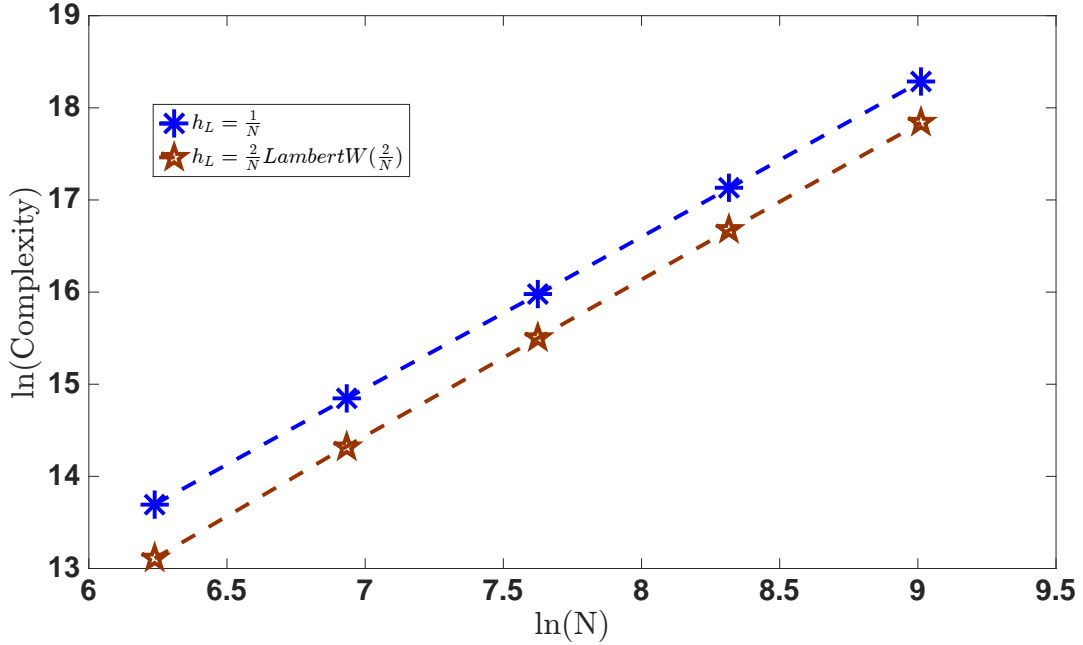


Figure 4: Complexity comparison of unbiased multilevel Monte Carlo tau-leaping when $h_L = \frac{1}{N}$ and $h_L = \frac{2}{N} \text{LambertW}(\frac{2}{N})$, with $\varepsilon_N = N^{-\frac{5}{4}}$.

tion for *whether* and *under what circumstances* approximation methods offer computational benefit has proved elusive. Focusing on the classical scaling, we note that a useful analysis must resolve two issues:

- (1) Computational complexity is most relevant for “large” problems, where many events take place. However, as the system size grows the problem converges to a simpler, deterministic limit that is cheap to solve.
- (2) On a fixed problem, in the traditional numerical analysis setting where mesh size tends to zero, discretization methods become arbitrarily more expensive than exact simulation because the exact solution is piecewise constant.

In this work, we offer what we believe to be the first rigorous complexity analysis that allows for systematic comparison of simulation methods. The results, summarized in Table 1, apply under the classical scaling for a family of problems parametrized by the system size, N , with accuracy requirement $N^{-\alpha}$. In this regime, we can study performance on “large” problems when fluctuations are still relevant.

A simple conclusion from our analysis is that standard tau-leaping does offer a concrete advantage over exact simulation when the accuracy requirement is not too high, $\alpha < 1$; see the first two rows of Table 1. Also, “second order” midpoint or trapezoidal tau-leaping improves on exact simulation for $\alpha < 2$; row three of Table 1. Furthermore, in this framework, we were able to analyze the use of a diffusion, or Langevin, approximation and the multilevel Monte Carlo versions of tau-leaping and diffusion simulation. Our overall conclusion is that some form of tau-leaping is always worthwhile. For low accuracy ($\alpha < 2/3$), second order

tau-leaping with standard Monte Carlo is the most efficient of the methods considered. At higher accuracy requirements, $\alpha > 2/3$, multilevel Monte Carlo tau-leaping is joint-best. Moreover, for all $\alpha > 1$ an unbiased version of multilevel Monte Carlo tau-leaping shares the lowest complexity level, making it our method of choice for high accuracy.

Possibilities for further research along the lines opened up by this work include:

- analyzing other methods within this framework, for example, (a) multilevel Monte Carlo for the diffusion approximation using discretization methods customized for small noise systems, or (b) methods that tackle the Chemical Master Equation directly using large scale deterministic ODE technology [22, 23],
- coupling the required accuracy to the system size in other scaling regimes, for example, to study specific problem classes with multiscale structure [9].

References

- [1] D. F. ANDERSON, *A modified next reaction method for simulating chemical systems with time dependent propensities and delays*, J. Chem. Phys., 127 (2007), p. 214107.
- [2] D. F. ANDERSON, A. GANGULY, AND T. G. KURTZ, *Error analysis of tau-leap simulation methods*, Annals of Applied Probability, 21 (2011), pp. 2226–2262.
- [3] D. F. ANDERSON AND D. J. HIGHAM, *Multilevel Monte-Carlo for continuous time Markov chains, with applications in biochemical kinetics*, SIAM: Multiscale Modeling and Simulation, 10 (2012), pp. 146–179.
- [4] D. F. ANDERSON, D. J. HIGHAM, AND Y. SUN, *Complexity analysis of multilevel Monte Carlo tau-leaping*, SIAM J. Numer. Anal., 52 (2014), pp. 3106–3127.
- [5] D. F. ANDERSON, D. J. HIGHAM, AND Y. SUN, *Multilevel Monte Carlo for stochastic differential equations with small noise*, arXiv preprint arXiv:1412.3039, (2014).
- [6] D. F. ANDERSON AND M. KOYAMA, *Weak error analysis of numerical methods for stochastic models of population processes*, SIAM: Multiscale Modeling and Simulation, 10 (2012), pp. 1493–1524.
- [7] D. F. ANDERSON AND T. G. KURTZ, *Continuous time Markov chain models for chemical reaction networks*, in Design and Analysis of Biomolecular Circuits: Engineering Approaches to Systems and Synthetic Biology, H. Koepl, D. Densmore, G. Setti, and M. di Bernardo, eds., Springer, 2011, pp. 3–42.
- [8] —, *Stochastic analysis of biochemical systems*, vol. 1.2 of Stochastics in Biological Systems, Springer International Publishing, Switzerland, 1 ed., 2015.
- [9] K. BALL, T. G. KURTZ, L. POPOVIC, AND G. REMPALA, *Asymptotic analysis of multiscale approximations to reaction networks*, Annals of Applied Probability, 16 (2006), pp. 1925–1961.

- [10] A. F. BARTHOLOMAY, *Stochastic models for chemical reactions. I. Theory of the unimolecular reaction process*, Bull. Math. Biophys., 20 (1958), pp. 175–190.
- [11] —, *Stochastic models for chemical reactions. II. The unimolecular rate constant*, Bull. Math. Biophys., 21 (1959), pp. 363–373.
- [12] M. DELBRÜCK, *Statistical fluctuations in autocatalytic reactions*, The Journal of Chemical Physics, 8 (1940), pp. 120–124.
- [13] C. W. GARDINER, *Handbook of Stochastic Methods: for Physics, Chemistry and the Natural Sciences*, Springer, Berlin, 2002.
- [14] M. GIBSON AND J. BRUCK, *Efficient exact stochastic simulation of chemical systems with many species and many channels*, J. Phys. Chem. A, 105 (2000), pp. 1876–1889.
- [15] M. GILES, *Improved multilevel Monte Carlo convergence using the Milstein scheme*, in Monte Carlo and Quasi-Monte Carlo Methods 2006, A. Keller, S. Heinrich, and H. Niederreiter, eds., Springer-Verlag, 2007, pp. 343–358.
- [16] M. GILES, *Multilevel Monte Carlo path simulation*, Operations Research, 56 (2008), pp. 607–617.
- [17] D. T. GILLESPIE, *A general method for numerically simulating the stochastic time evolution of coupled chemical reactions*, J. Comput. Phys., 22 (1976), pp. 403–434.
- [18] —, *Exact stochastic simulation of coupled chemical reactions*, J. Phys. Chem., 81 (1977), pp. 2340–2361.
- [19] —, *Approximate accelerated simulation of chemically reacting systems*, J. Chem. Phys., 115 (2001), pp. 1716–1733.
- [20] M. HUTZENTHALER, A. JENTZEN, AND P. E. KLOEDEN, *Strong and weak divergence in finite time of Euler’s method for stochastic differential equations with non-globally Lipschitz continuous coefficients*, Proceedings of the Royal Society A, 467 (2011), pp. 1563–1576.
- [21] M. HUTZENTHALER, A. JENTZEN, AND P. E. KLOEDEN, *Divergence of the multilevel Monte Carlo Euler method for nonlinear stochastic differential equations*, Ann. Appl. Prob., 23 (2013), pp. 1913–1966.
- [22] T. JAHNKE, *On reduced models for the chemical master equation*, SIAM: Multiscale Modeling and Simulation, 9 (2011), pp. 1646–1676.
- [23] V. KAZEEV, M. KHAMMASH, M. NIP, AND C. SCHWAB, *Direct solution of the chemical master equation using quantized tensor trains*, PLOS Computational Biology, (2014).
- [24] P. E. KLOEDEN AND E. PLATEN, *Numerical Solution of Stochastic Differential Equations*, vol. 23 of Applications of Mathematics (New York), Springer-Verlag, Berlin, 1992.

- [25] T. G. KURTZ, *The relationship between stochastic and deterministic models for chemical reactions*, J. Chem. Phys., 57 (1972), pp. 2976–2978.
- [26] ———, *Strong approximation theorems for density dependent Markov chains*, Stoch. Proc. Appl., 6 (1978), pp. 223–240.
- [27] ———, *Representation and Approximation of Counting Processes*, vol. 42 of Advances in filtering and optimal stochastic control, Springer, Berlin, 1982.
- [28] D. A. MCQUARRIE, *Stochastic approach to chemical kinetics*, J. Appl. Prob., 4 (1967), pp. 413–478.
- [29] E. RENSHAW, *Stochastic Population Processes*, Oxford University Press, Oxford, 2011.
- [30] S. M. ROSS, *Simulation*, Academic Press, Burlington, MA, fourth ed., 2006.
- [31] D. J. WILKINSON, *Stochastic Modelling for Systems Biology*, Chapman and Hall/CRC Press, second ed., 2011.